Using Film–Trope Connections for Clustering Similar Movies

Cecilia Äijälä

Minor Subject Thesis
UNIVERSITY OF HELSINKI
Department of Computer Science

Helsinki, April 29, 2020

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Computer Science

Tekijä — Författare — Author

Cecilia Äijälä

Työn nimi — Arbetets titel — Title

Using Film–Trope Connections for Clustering Similar Movies

Oppiaine — Läroämne — Subject

Computer Science

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Minor Subject Thesis	April 29, 2020	33

Tiivistelmä — Referat — Abstract

Tropes are storytelling devices or conventions that can be found in storytelling media, for example in movies. DBTropes is an RDF-dataset with media-trope connections extracted from Tv Tropes, which is a community-edited wiki listing tropes for various creative works.

This study investigates whether the tropes of films can be used to cluster similar movies. First, we extracted film-trope connections from the DBTropes dataset. We then took four samples from the dataset, three for clustering films, and one for clustering tropes.

We used the film–trope connections to calculate *euclidean* and *cosine* distances between movies and for the last sample between tropes. Then we clustered the samples with hierarchical clustering using *complete* and *ward* linkage.

Our results show that hierarchical clustering can group similar films together using this dataset. For calculating distances the *cosine* distance method works significantly better than *euclidean* distance. Both hierarchical clustering methods *complete* and *ward* work well. It depends on the chosen sample, which of them results in a clearer and more interpretable output. We conclude that the data works well for clustering similar films or similar tropes.

Avainsanat — Nyckelord — Keywords

Film, Movie, Trope, Hierarchical Clustering, TV Trope, DBTrope,

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1 Introduction **1**

2 DBTropes data **2**
2.1 The DBTropes dataset 2
2.2 Extracting the film–trope connections 4
2.3 Data samples 5

3 Methods **8**
3.1 Pairwise distance 8
3.2 Hierarchical clustering 9

4 Clustering Tropes and Movies **10**
4.1 Clustering Movies in the *Most Tropes Sample* 10
4.2 Clustering Movies in the IMDB Categories Sample 15
4.3 Clustering Movies in the Guinness Sample 18
4.4 Clustering Tropes in the Most Films Sample 21

5 Conclusions **24**

References **25**

Appendices **27**

A Sample Films **27**

B Trope Info **31**

1 Introduction

Movie recommendation is a modern area of research, as good recommendations are much in demand in the world today. Finding out which movies are similar to each other is not an easy task because movies usually belong to more than one genre. Even movies belonging to the same genres can differ a lot from each other.

The aim of this project is to look at data on film–trope connections to see if the tropes of films can be used to group similar movies. Tropes are storytelling devices or conventions. They can, for example, be plot tricks, narrative structures, or character types [2]. Three-Act-Structure is an example of a narrative structure trope and Gentleman-Thief is an example of a character type trope. The community-maintained wiki Tv Tropes[1] collects and classifies tropes and connects them to films and other storytelling media like books, tv-serie's, and manga.

In this study we are going to use the RDF-dataset called DBTropes[2] by Kiesel [3]. DBTropes is an online wrapper for Tv Tropes, and it contains media–trope connections extracted from the site. The latest version was created in July 2016.

We first extract the film–trope connections from the dataset using the SPARQ query-language. Then we take four samples from the dataset to use in the rest of the study. three for clustering films and one for clustering tropes. We call the samples *Most Tropes*, *IMDB Categories*, *Guinness* and *Most Films*. Next, we calculate the distance between films, and for the fourth sample between tropes. We use *euclidean* and *cosine* distance to calculate distance from the film–trope connections, and cluster the samples using hierarchical agglomerative clustering. For this we use the linkage functions *cosine* and *ward*. Thus we end up with four combinations, namely: *euclidean–complete*, *euclidean–ward*, *cosine–complete* and *cosine–ward*, that we will use to analyze the samples.

The remainder of the paper is organized as follows. In Chapter 2 we describe the DBTropes dataset and related work done with it, as well as the process of extracting the film–trope connections from it and the samples we took from the extracted dataset for the clustering study. In Chapter 3 we go through the methods used for clustering. The results of the clustering for all the four samples will be shown in Chapter 4, where we will also do some short analysis of the clusters. Lastly, we conclude the paper with Chapter 5 describing our conclusions and some possible future lines of work.

---

[1] `tvtropes.org`
[2] `dbtropes.org`

2 DBTropes data

Let us take a look at the DBTropes dataset in this chapter. We will first look at the dataset and the work that had been done with it earlier (Chapter 2.1), and then we will go over the methods of extracting the film–trope connections from the dataset for this study (Chapter 2.2). Finally, we will look at the samples taken from the extracted dataset that we will use for the rest of the study (Chapter 2.3).

2.1 The DBTropes dataset

The latest downloadable dump of the RDF-dataset DBTropes was generated 1st of July 2016, and it contains 21 057 602 RDF statements. The data is in n-triples format and has a size of 4.7 GB. In the n-triples format, the data is in plain text, and each line contains a complete RDF statement. The statement consists of four parts that are separated from each other with whitespace; the subject, the predicate, the object, and a full stop. The subject of a statement may be a blank or a URI, predicates must be URIs, and objects may be a blank node, a URI or a literal [17]. In this dataset, both media and tropes are represented by URIs and can be either subjects or objects. Instead of a URI, the object can sometimes be a comment or some other text string. An example of the data containing five statements from the dataset can be seen in Figure 1.

As we can see in Figure 1, films and other works of fiction are represented by URIs like; *http://dbtropes.org/resource/<mediaCategory>/<resourseName>* where *<mediaCategory>* is e.g. *Film* or *WebComic*, and *<resourseName>* is the name of that resource. The tropes are almost the same as the films. The difference is that it has the word *Main* at the place of the <mediaCategory>. This follows the structure of the URLs of the Tv Trope webpage. If the URL of a Tv Trope webpage had an error in its spelling, that error will also be present in the DBTrope dataset. Sometimes the URI has an additional ID like *int_<IdCode>* at the end so that other RDF statements can refer to a specific connection.

The media and the tropes are connected with many different predicates. The predicates are URI:s but in this work, we will name the predicate by its last component. The most common predicates are *type* and *hasFeature* with the media as the subject and the trope as an object. When comparing the dataset to the webpage it seems that these connections correlate with the current data on the website. Sometimes the *hasFeature* connections are found only on a tropes webpage linking to a film as an example of the trope and not on the page of the film. Then again, some of the *hasFeature* connections look the same on the webpage as the type connections. The inconsistency in the tags in the dataset is probably due to changes that have happened on the constantly changing webpage after the latest DBTropes dump 3 years earlier,

```
<http://dbtropes.org/resource/Film/TheFrighteners>
<http://skipforward.net/skipforward/resource/seeder/
skipinions/hasFeature> <http://dbtropes.org/resource/
Main/BodyCountCompetition/int_419065b3> .

<http://dbtropes.org/resource/Webcomic/TheZombieHunters>
<http://skipforward.net/skipforward/resource/seeder/
skipinions/hasFeature>  <http://dbtropes.org/resource/
Main/BodyCountCompetition/int_8d853d96> .

<http://dbtropes.org/resource/Main/BodyCountCompetition/
int_5dc85d5f>  <http://www.w3.org/2000/01/rdf-schema#comment>
"Gimli and Legolas during the Battle of the Hornburg in The
Lord of the Rings, occasionally asking other characters if
they've seen Legolas/Gimli to update the score. Gimli wins
by a point, but Legolas doesn't begrudge it, because they
were each just glad the other was alive."@en .

<http://dbtropes.org/resource/Main/BodyCountCompetition/
int_8d853d96>  <http://www.w3.org/1999/02/
22-rdf-syntax-ns#type>  <http://dbtropes.org/resource/
Main/BodyCountCompetition> .

<http://dbtropes.org/resource/Main/BodyCountCompetition/
int_e03a0291>  <http://skipforward.net/skipforward/resource/
seeder/skipinions/featureApplicability>
"1.0"\textasciicircum\textasciicircum<http://
www.w3.org/2001/XMLSchema#> .
```

Figure 1: An example from the DBTropes dataset

as well as inconsistent updates from users as Tv Tropes is a community updated page. There were also predicates labeled as *processingUnknown* with tropes as subjects and films as objects and vice versa. According to the DBTropes webpage, *processingUnknown* is "indicating referenced pages whose type is not known yet". When checking a subset of these they were as consistent as the other tags, like *type* and *hasFeature*, with the current webpage and they contained altogether 2595 films that were not connected to any tropes with *type* or *hasFeature*. There were also other predicates than those mentioned above, that connect films and tropes in the data, but we ignored these connections (Chapter 2.2).

Kiesel and Grimmes [9]) made a test on the recall and precision of the datasets extraction process in 2010 to see how well Tv Tropes were extracted to DBTropes . The recall and precision of tropes are about 83% and 95.3% and for films about 89.7% and 96.2% respectively. They also state that most of the errors are "due to missing primary information in the original wiki".

Even though the data is a snapshot of a community updated, constantly changing wiki, these results give good confidence on at least the extraction process made from Tv Tropes to DBTropes.

Kiesel et. al. [11, 10] used the DBTropes data in Skipforward[3], an ontology-based distributed annotation system, where people can e.g. like and dislike items, and which gives users recommendations of similar items. The data has also been used in other research on movie recommender algorithms. Yang [18] used semantic data, among others the DBTropes dataset, to improve recommender systems based on Slope One, which is a family of algorithms proposed for collaborative filtering. Peska [14, 15] used the data in two papers where he extended the MovieLens 1M dataset [8] using DBTropes and IMDB movie features to extend the film profiles, and ZIP code statistics to extend user profiles. In the first paper, Peska [14] proposed a multiple content alignment extension to the Bayesian Personalized Ranking Matrix Factorization (BPR-MCA) and evaluated that using the extended MovieLens 1M dataset. The BPR-MCA provides a significant improvement over standard BPR [16]. The results also pointed out the importance of using informative content-based attributes. In the second paper [15], Peska proposes a Content-Aligned Bayesian Personalized Ranking Matrix Factorisation (CABPR) to extend the standard BPR, which is a significant extension to the BPR-MCA. The dataset has also been used for interpreting latent factors in recommender systems that use matrix factorization. Datta et. al. [5] used the data by integrating auxiliary features from DBTropes and IMDB to the MovieLens dataset. García-Ortega et. al. [6] extracted the film–trope connections from DBTropes to a dataset called PicTropes. Later they made an application called tropescraper[4] to scrape trope–film connections from Tv Tropes. They used the software to get more up-to-date data for their film trope rating optimizer called StarTroper [7]. The idea of StarTroper is to synthesize sets of tropes that maximize the potential rating of a film that conforms to them.

2.2 Extracting the film–trope connections

In this project, our goal is to examine connections between films and tropes. Let us define films as those works of fiction with URIs containing the *mediaCategory Film* or *Disney* as well as all misspelled versions of them. The *Disney* category contains very similar films to the *Film* category, and it is the only western film producer with its own tag. On the other hand, we left out movie containing categories e.g. *Bollywood* and *Anime*, as they are quite different, and for example the *Anime* category makes no distinction between films and series. All tropes with at least one connection to a film were saved to the extracted dataset.

[3] `http://skipforward.opendfki.de/wiki/WikiStart`
[4] `https://github.com/raiben/tropescraper`

To extract the desired data we first split the dataset into 4 parts to decrease running times and memory requirements. We did this with a simple command-line file splitting operation. Splitting the data at the end of a random row did not break the data and the results did not change because the n-triples format is valid row by row. We then queried the data with the RDF-query language called SPARQL. This was done using the Apache Jena SPARQL-query command-line tool [1]. We first queried the data for all RDF-statements having a film or a trope as subject or object. Then we extracted the connections with the desired datatypes. We decided to keep the statements with film as a subject and trope as an object connected by *type* and *hasFeature* as well as those linked by the predicate *processingUnknown* both with film as an object and trope as a subject and vice versa. In the cases when the URI of a film or trope contained an ID at the end that ID was removed so that all connections referring to the same film or trope have the same URI.

We saved the extracted data in CSV-format, where each row contains a film URI, a trope URI and a tag telling with which predicate they were connected in the dataset. Some film–trope pairs were connected to each other by both *type* and *hasFeature*. In these cases we saved the *type* connection and discarded the *hasFeature* connection. To save the data in a more compact form we gave all film and trope URI:s an ID. The uploaded version of the data has only the info of *trope-id*, *film-id* and *connection-id*. Additionally, there are separate metadata files for both films and tropes, with the URI of the film/trope, their ID in the connection file, and the URL to their Tv Trope page as given in the data. For the connections, there is a metadata file with *connection-id* and the name of the connection.

The statistics of the extracted dataset are shown in Table 1. The dataset contains 9441 films and 22977 tropes. The total amount of film–trope pairs is 379124 from which 267427 are with the *type* tag, 149349 are with the *hasFeature* tag and 22663 are with the *processingUnknown* tags.

The range of the number of tropes in a film and films with a trope is large (Figure 2). The largest number of tropes in a film is 1148. The average amount is, however, much smaller (mean 40.157, median 19). There are 1068 films with only one trope. The amount of films with a specific trope follows a similar distribution as the number of tropes in a specific film (Figure 2) with the most used trope existing in 1540 films. The average is again much smaller (mean 16.5, median 7). Of all the 22977 tropes in the dataset 3459 can be found in only one of the films.

2.3 Data samples

In this project, we aimed to see if the DB-tropes film–trope pairs (Chapter 2.2) could be used to classify similar films. To do this we took 3 different samples from the data.

Table 1: Statistics on the number of tropes and films. PU stands for the *processingUnknown* tag.

Number of Films	9441
Number of Tropes	22977
Rows of Data	379124
of which rows with connection:	
Type	267427
HasFeature	149349
PU Trope to Film	16962
PU Film to Trope	5701

	Films Per Trope	Tropes Per Film
Min	1	1
Max	1540	1148
Mean	16.5	40.157
Median	7	19

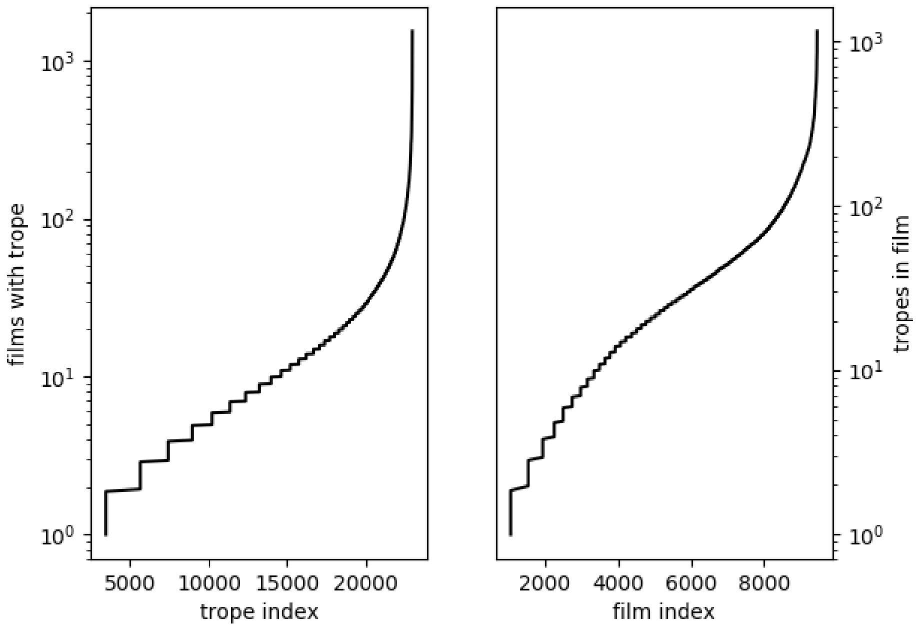

Figure 2: Distribution of the amount of films with a trope (left) and tropes in a film (right).

We used small subsets (50–60 films) to analyze the resulting hierarchical tree diagrams called dendrograms. We generated the first dataset by taking 50 films with the most tropes in them and keeping all tropes that were in at least 2 of the selected films. This dataset has 6127 tropes and 24854 connections altogether.

As the films with the most tropes are primarily action movies, we created two other subsets that aim to be a bit more balanced, even though it is hard to do class distinctions for films as they usually belong to multiple genres. We created the first of these by taking the top 10 films ordered by votes from five different film categories on the webpage IMDB (Internet Movie Database)[5]. The genres *comedy*, *horror*, *romance*, *mystery* and *action* were selected. Tropes were again chosen by taking all the tropes that were in at least two films. The sample has 50 films, 2330 tropes, and 7143 connections.

Table 2: Statistics on films and tropes in each of the samples. *Most tropes* contain the 50 films having the most tropes, *IMDB categories* contain 10 films each from 5 different IMDB categories, *Guinness* contains 12 categories with 5 films each from the Guinness' top 100 films and *Most films* contain the 50 tropes in most films.

	Most tropes	IMDB categories	Guinness	Most films
Films	50	50	60	4390
Tropes	6172	2363	1923	50
Connections	24854	7272	6027	25256
Tropes/Film				
Min	315	1	6	2
Max	892	534	332	31
Mean	497.08	145.44	100.45	5.75
Median	454	121	74	4
Films/Trope				
Min	2	2	2	312
Max	47	25	29	1484
Mean	4.056	3.077	3.13	505.12
Median	3	2	2	446

We generated the second of these additional samples by taking the Guinness' top 100 films list [4] that was published in Guinness book of films in 1999 and consisted of 20 genre categories of films with 5 films each. From this list, we picked those categories, where all five films can be found in the dataset, and where all films contain at least one trope that is found in more than one film. This left us 12 categories with a total of 60 films altogether;

[5] `https://www.imdb.com/feature/genre/` (accessed 5.12.2019)

action, *comedy*, *romance*, *science fiction*, *disaster*, *spy*, *drama*, *swashbuckler*, *cult*, *fantasy*, *gangster* and *horror*. The sample contains 1878 tropes and 5827 connections all in all. Appendix A contains all films in these samples, the number of tropes they have in each sample, and all the film genres they belong to according to IMDB.

Additionally, we made a sample of the 50 most common film tropes in the dataset, to use when trying to cluster tropes using film connections. This sample has 4390 films and 25256 connections altogether. The statistics on all the subsets can be seen in Table 2.

3 Methods

To analyze the data we first calculated the distance between the films (Chapter 3.1) and applied hierarchical clustering (Chapter 3.2) to see what movies have similar tropes. The aim was to see if similar films are close to each other in the final results (shown in Chapter 4). Also, one additional subset was taken to see what kind of results we would get when trying to cluster tropes by their film connections instead. For this, the methods were otherwise identical to the film clustering, except we clustered tropes using films.

3.1 Pairwise distance

First, we represent each movie with a binary feature vector. We do this by assigning an index from 1 to n to each trope, and for each film, an index from 1 to m. Thus each film is represented by a vector of length n. We set $X_{i,j} = 1$ if the film f_i had the trope t_j, and $X_{i,j} = 0$ otherwise. Thus we get the feature matrix $\mathbf{X}$ which has the size of $m \times n$. An example illustrating the resulting matrix can be seen in Figure 3.

Using this matrix, we can then calculate the distances between films. Doing this we end up with a distance matrix $\mathbf{D}$ of size $m \times m$, where $D_{a,b}$ is the distance between the films a and b. We did this by using two different metrics; l_2 and *cosine distance*.

$$
\begin{array}{cccccc}
 & t_1 & t_2 & t_3 & \dots & t_n \\
f_1 & 1 & 0 & 0 & \dots & 1 \\
f_2 & 0 & 0 & 1 & \dots & 0 \\
\dots & \dots & \dots & \dots & \dots & \dots \\
f_m & 0 & 1 & 0 & \dots & 0
\end{array}
$$

Figure 3: Example illustrating a matrix $\mathbf{X}$ of film vectors, where f stands for films and t stands for tropes. In this example film f_1 has tropes t_1 and t_n, film f_2 has trope t_3, and film f_m has trope t_2.

In this work, we consider two distances, l_2 is the euclidean distance between 2 vectors,

$$d_{l2}(a, b) = ||a - b||_2,$$

where a and b are 2 row vectors from matrix $\mathbf{X}$. The distance is calculated for each pair of row vectors. *Cosine distance* is the complement of the l_2-normalized dot product of 2 row vectors,

$$d_{cosine}(a, b) = 1 - \frac{ab^T}{|a||b|},$$

and it is similarly calculated for all possible row vector pairs from matrix $\mathbf{X}$. The calculations were done using the python method *pairwise_distances*, from the *sklearn.metrics.pairwise* submodule [13].

3.2 Hierarchical clustering

Hierarchical clustering is an unsupervised cluster analysis method for clustering unlabeled data points. A typical goal is to construct a tree where the leaves are singular points. Hierarchical clustering methods can be divided into two categories, agglomerative and divisive.

In agglomerative clustering [12], the points are clustered from the bottom up starting from individual points that are each treated as singleton clusters. Thus at the start, there are M clusters, where M is the number of data points. Next, we form a new cluster by joining the two closest clusters resulting in $M - 1$ clusters. This is repeated by always combining the 2 closest clusters until a single cluster is left. We can visualize the clustering with hierarchical tree diagrams, called dendrograms.

The divisive hierarchical clustering algorithm is similar except we go top-down starting by treating all the points as one big cluster and then repeatedly splitting the most appropriate cluster. There are different methods to calculate the distance between the clusters used when choosing the two closest clusters [12].

In agglomerative clustering, we need new distances for the formed clusters in each step, and not only the distances between the singleton clusters. How these new cluster distances are calculated depends on the clustering linkage function used. In this study we use the linkage functions *complete* and *ward*.

The function *complete* uses the *farthest point* method,

$$d_{complete}(U, V) = max(d(u, v)).$$

The new distance $d_{complete}(U, V)$ is the farthest distance between any film u in cluster U to any film v in cluster V, where U and V are clusters at the current step of the agglomerate clustering algorithm.

The linkage function *ward* is used by the *ward variance minimization algorithm*, and is defined as

$$d_{ward}(U,V) = \sqrt{\frac{|V|+|S|}{T}d(V,S)^2 + \frac{|V|+|R|}{T}d(V,R)^2 - \frac{|V|}{T}d(S,R)^2},$$

where $T = |V| + |S| + |R|$, and S and R are the clusters that were combined into the new cluster U at this step of the algorithm. The new distance $d_{ward}(U,V)$ is the new distance between a new cluster U and an unused cluster V.

We did the hierarchical clustering using the python method *linkage* from the *scipy.cluster.hierarchy* module [12]. The distance matrix $\mathbf{D}$ presented in Chapter 3.1 was used as input.

4 Clustering Tropes and Movies

We will now present the results of out clustering experiments. We used two distances, *euclidean* (l_2) and *cosine* (Chapter 3.1), and two ways of linking the clusters, ward and complete (Chapter 3.2). This gives us 4 combinations, namely: *euclidean–complete*, *euclidean–ward*, *cosine–complete* and *cosine–ward*. The samples *Most Tropes*, *IMDB categories* and *Guinnes* was used for clustering movies, and the sample *Most Films* was used for clustering tropes. The statistics of the samples can be seen in Chapter 2.3.

4.1 Clustering Movies in the *Most Tropes Sample*

The resulting dendrograms for the first sample, *Most Tropes*, are shown in Figures 4, 5, 6 and 7. We can see that children's cartoons are a separate cluster in all of these. In the *euclidean–complete* case this cluster also contains *Who Framed Roger Rabbit*, which is a partially animated movie. All of the combinations of methods are also able to cluster all the three X-Men movies together, and *The Dark Knight* and *The Dark Knight Rises* together. Additionally, when we used cosine distance the film *Batman* was clustered with the two newer Batman movies mentioned above. This is not true for the *euclidean* distance cases.

The cosine distance cases seem to be better at clustering films with their sequels. For example both *Captain America* films and The *Star Trek* films are next to each other in the cosine distance cases (Figures 6 and 7) but not in the *euclidean distance* cases (Figures 4 and 5). In addition, using cosine distance clusters *The Matrix* together with *Inception* and *Thor* together with *Guardians of the Galaxy*, which is good, as both *The Matrix* and *Inception* have layered realities and *Thor* and *Guardians of the Galaxy* are both Marvel movies. Such a match is missing when we use *euclidean* distance.

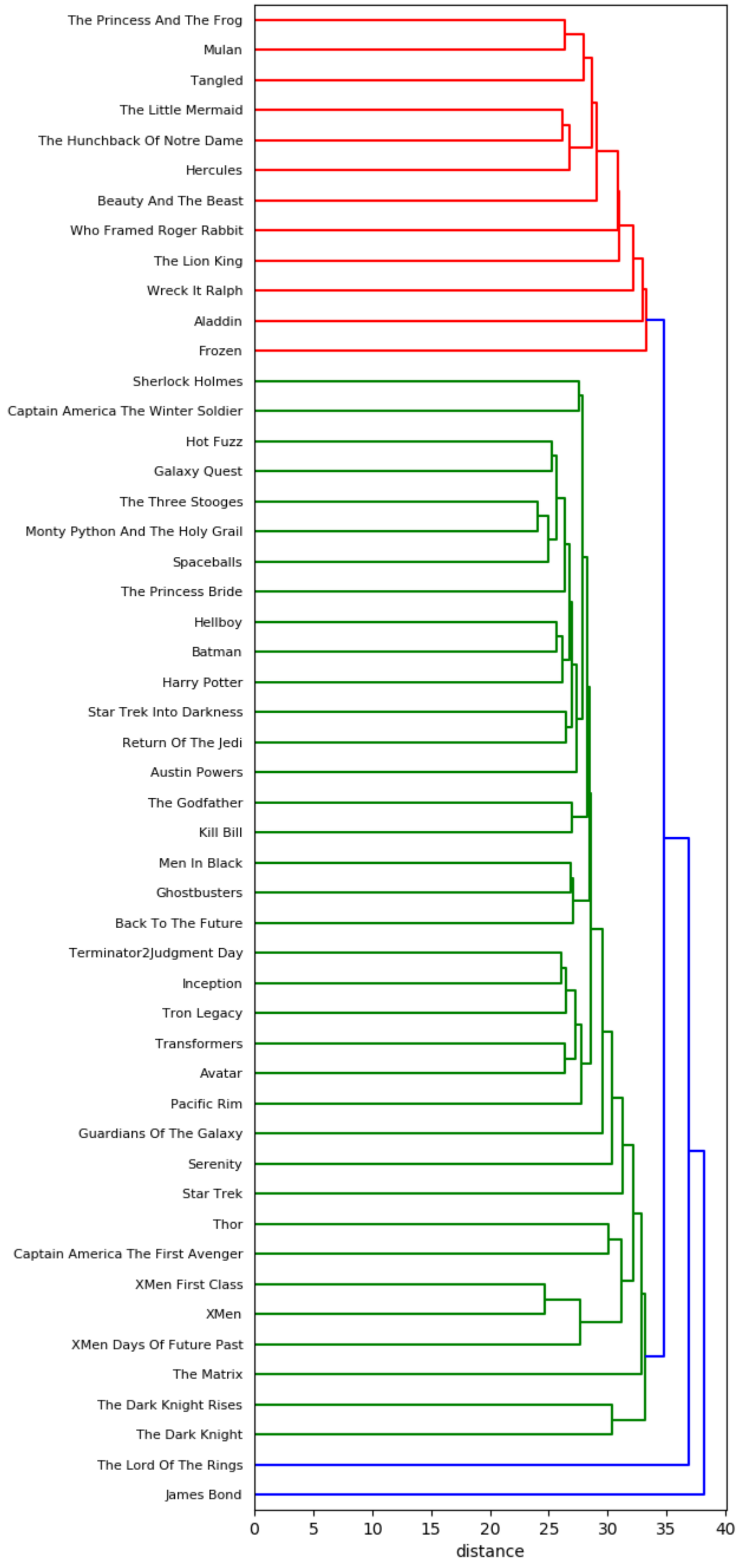

Figure 4: Dendrogram for the sample *Most Tropes* using the *euclidean* distance and *complete* linkage.

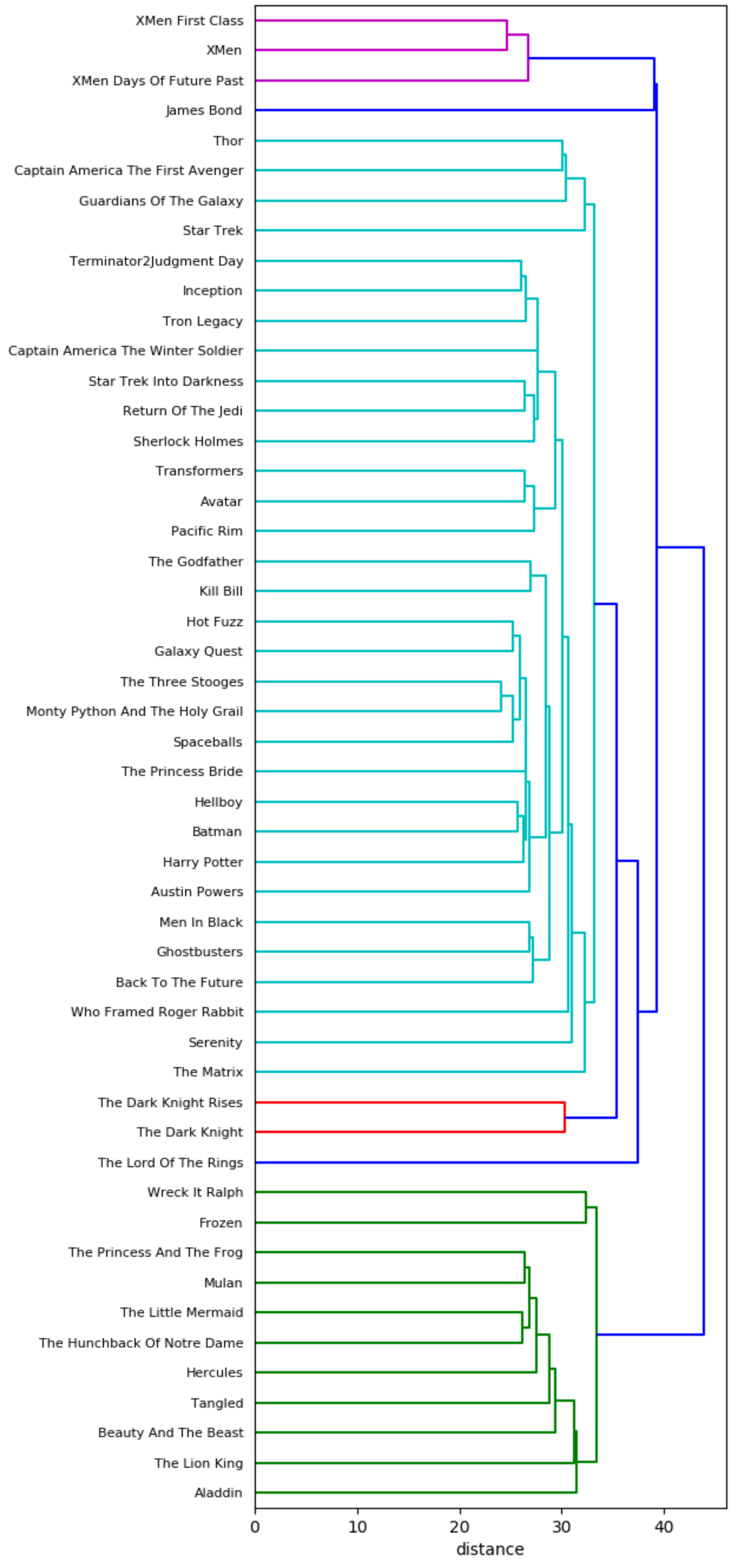

Figure 5: Dendrogram for the sample *Most Tropes* using the *euclidean* distance and *ward* linkage.

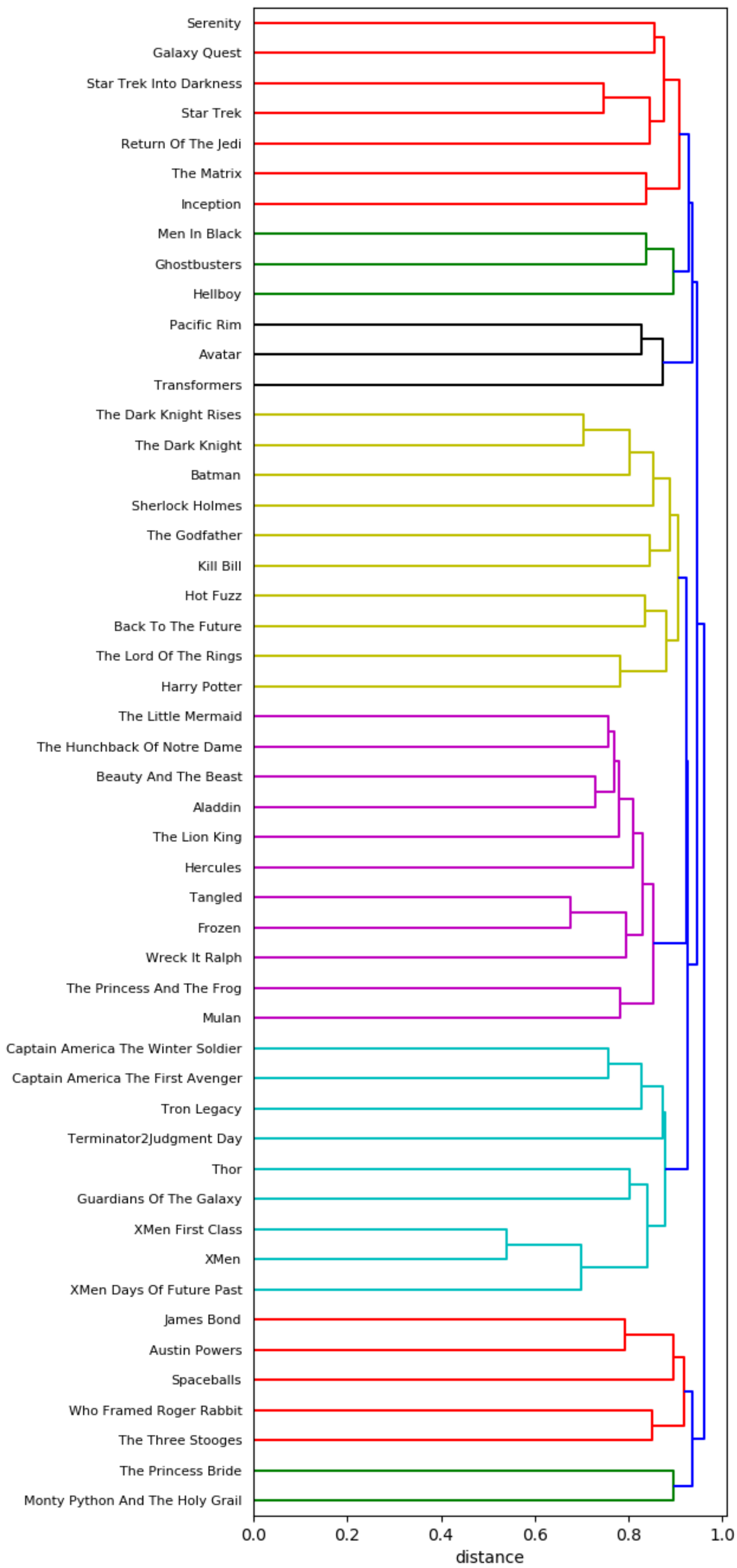

Figure 6: Dendrogram for the sample *Most Tropes* using the *cosine* distance and *complete* linkage.

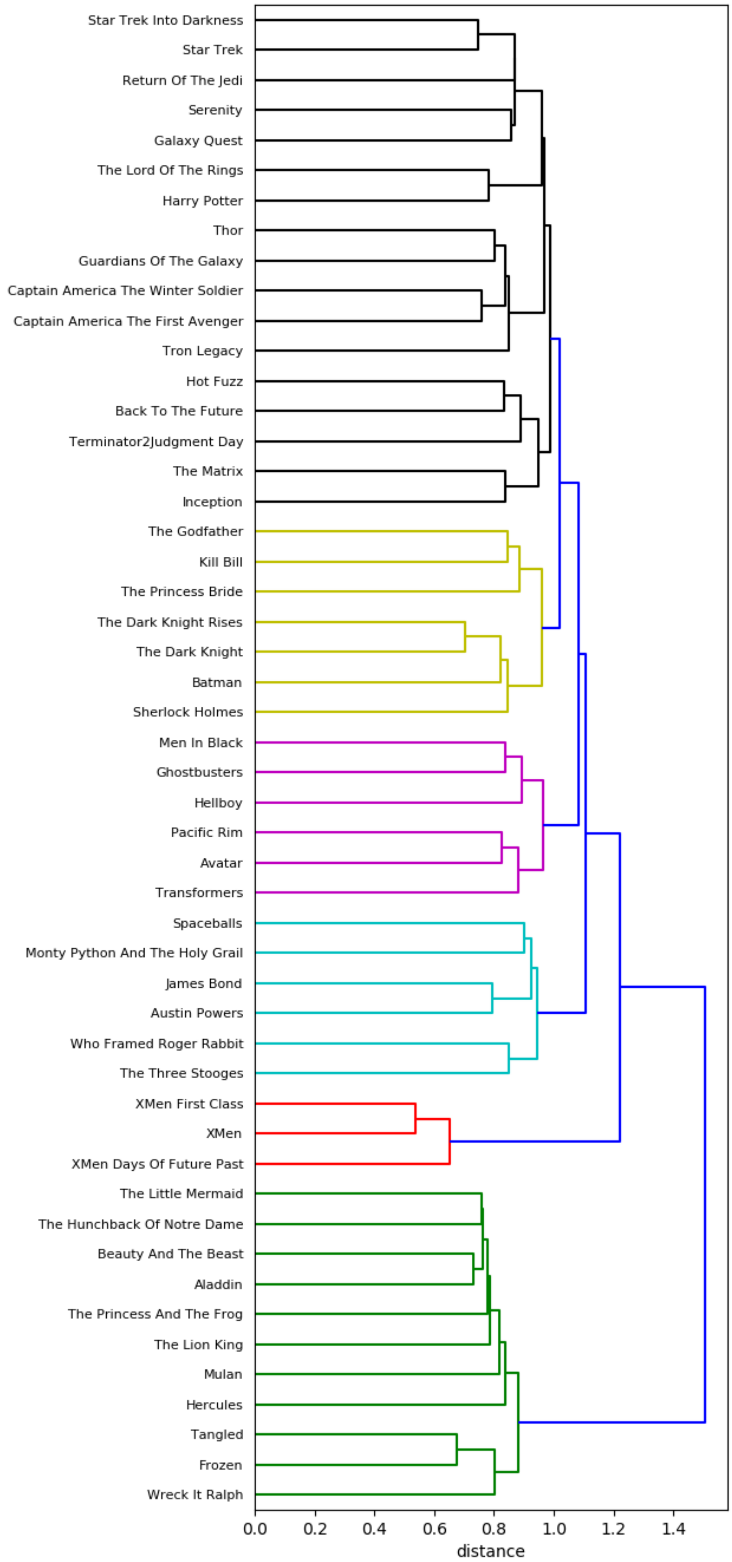

Figure 7: Dendrogram for the sample *Most Tropes* using the *cosine* distance and *ward* linkage.

Let us now take a closer look at the differences between the cosine distance cases. The first combined movies are mostly the same (Figures 6 and 7), as the distances are calculated with the same algorithm. The differences start to appear for larger clusters. For example, both cases have the films *Thor*, *Guardians of the Galaxy*, both *Captain Americas* and *Tron* in the same group but when using the *complete* linkage this group also contains the *X-Men* movies,while the same does not hold when using the *ward* linkage.

The Princess Bride is grouped with *Monty Python and the Holy Grail* in the *complete* case but with *Kill Bill* and *The Godfather* in the *ward* case. Of these both *Monty Python* and *The Princess Bride* are light-themed adventure and fantasy films, while *Kill Bill* and *The Godfather* are crime films (Appendix A). The film *James Bond* is in both cases connected to a group of light hearted movies, but that is because it is so similar to *Austin Powers* as they are both spy films, even though *Austin Powers* is more parody and *James Bond* is more serious.

In our view, the combination of complete linkage and cosine distance gives the most interpretable results for the sample *Most Tropes*.

4.2 Clustering Movies in the IMDB Categories Sample

For the second sample, *IMDB Categories*, we decided to only analyze the results from the instances using *cosine distance*. This was because the *cosine distance* was significantly better for the *Most Tropes* sample than the *euclidean* distance (see Chapter 4.1), and preliminary results showed that the same holds for the other samples. The results for *cosine–complete* can be seen in Figure 8 and those for *cosine–ward* can be seen in Figure 9.

We can see that many connections are the same in both instances. Both were able to group all 3 *Batman* movies. Both were also able to group the 4 zombie-themed horror movies *World War Z*, *28 Days Later*, *Zombieland*, and *Shaun of the Dead*. There are also some unexpected connections that both instances have. For example, *Gladiator*, which is an action, adventure, drama movie is connected to *Life Is Beautiful*, which is a comedy, drama, romance film, in both cases. When looking closer at this connection we think that this is because there are no better matches for *Life Is Beautiful*. Looking at the data we can see that it has 31 tropes in total, from which eight are common to *Guardians of the Galaxy* and *Shaun of the Dead* and seven with *Groundhog Day* and *Gladiator*. From these all but the *Gladiator* have better options to connect to. Another such idiosyncrasy is the fact that most comedy films got grouped together with the horror movie *Saw*. Looking at the number of tropes in all of these movies we see that they all have 1–2 tropes except *Deadpool* which has four. The maximum of common tropes with any film for *Deadpool* is one. Thus these connections are probably just due to lack of data on similarity for those films.

One notable difference between the two instances is that in the *cosine–ward*

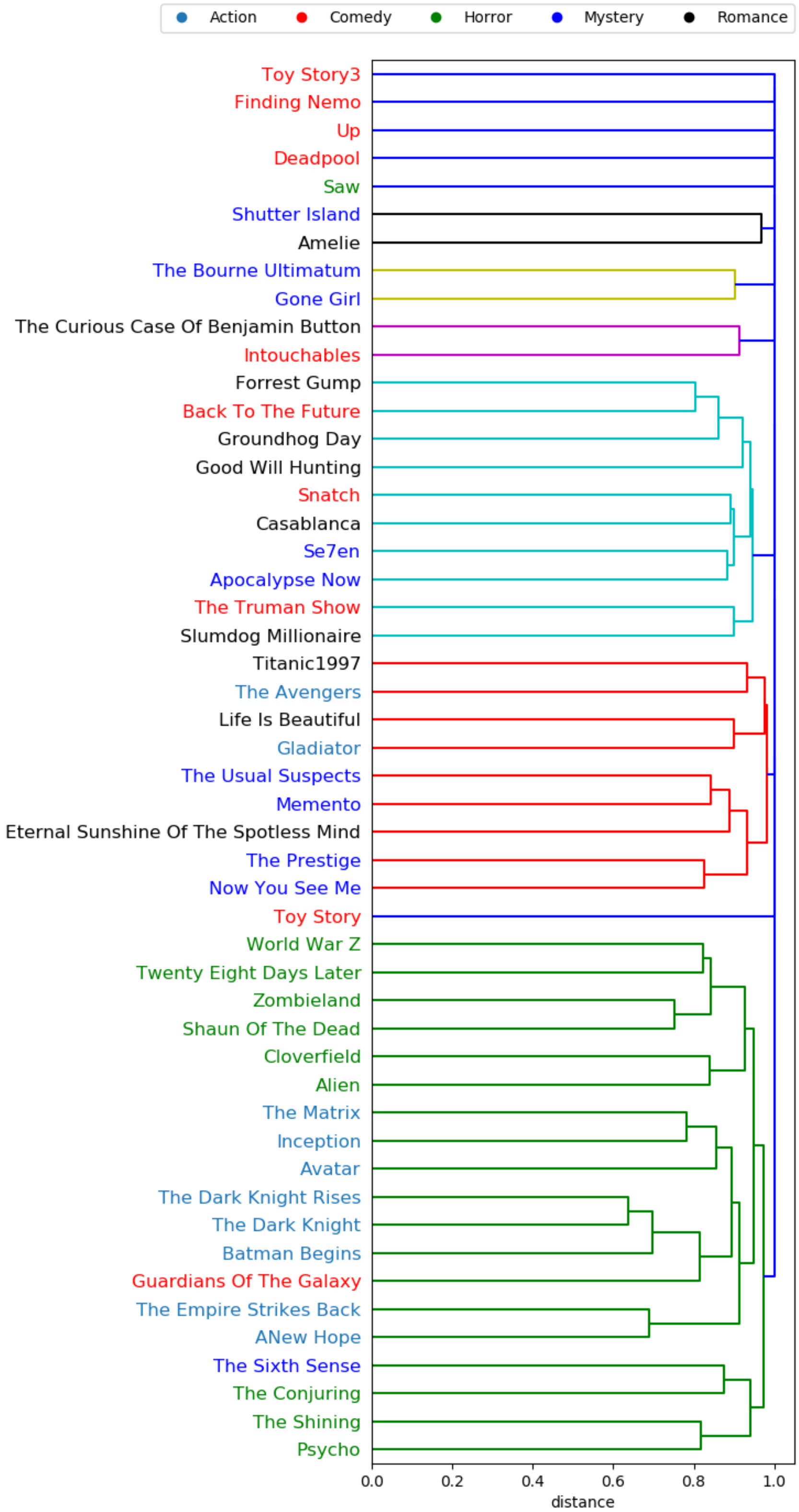

Figure 8: Dendrogram for the sample *IMDB categories* using the *cosine* distance and *complete* linkage. The legend and the film name colors indicate which genre the films belong to.

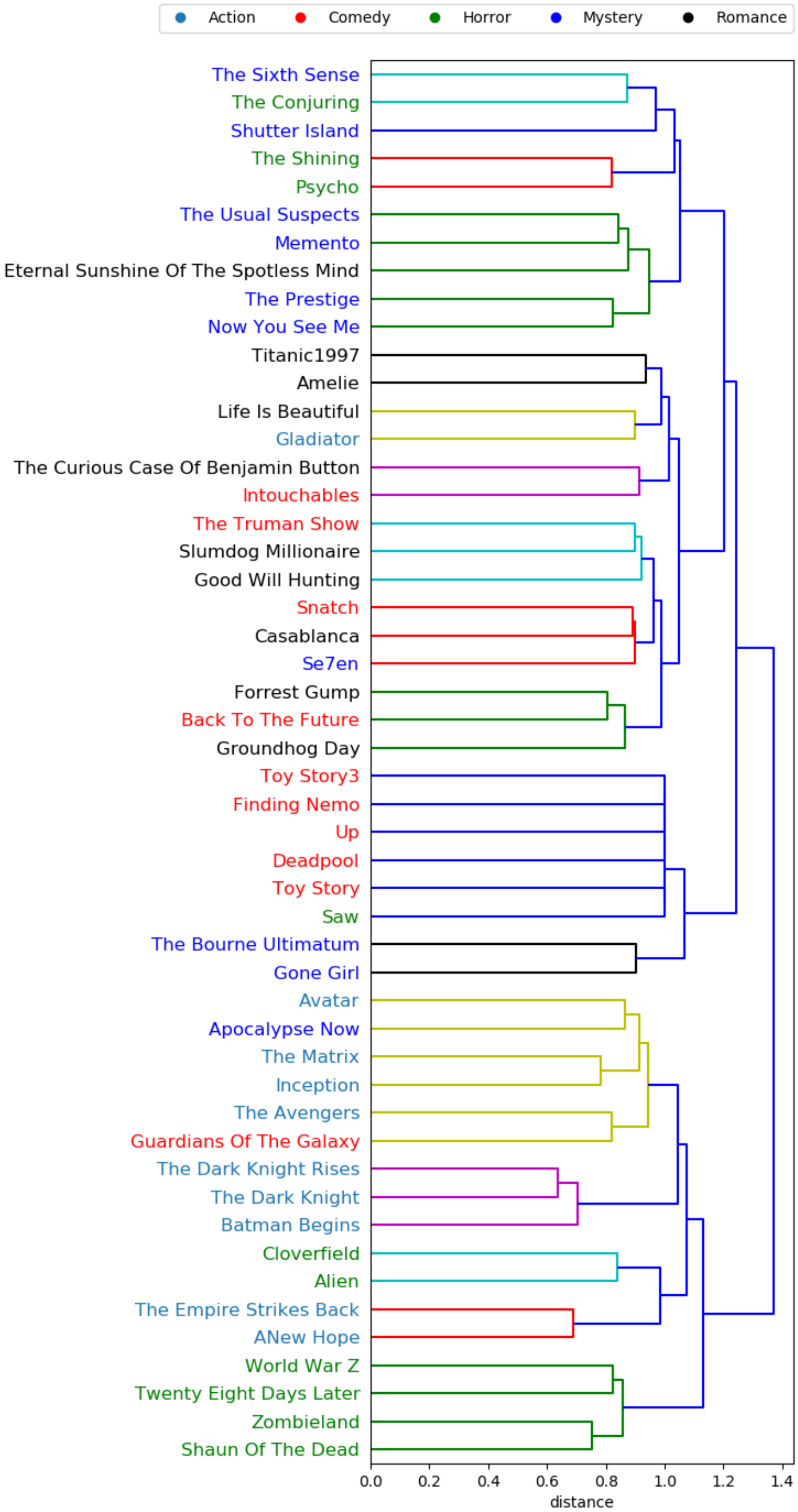

Figure 9: Dendrogram for the sample *IMDB Categories* using the *cosine* distance and *ward* linkage. The legend and the film name colors indicate which genre the films belong to.

case (Figure 9) the films *Titanic* and *Amélie* are next to each other while in the *cosine–complete* case *Titanic* is next to *The Avengers* and *Amélie* is next to *The Bourne Ultimatum*. *Cosine–ward* also seems to make deeper trees than *cosine–complete*.

All in all, the *cosine–ward* combination seems to perform well with this sample. There are some odd connections due to a small number of tropes or a lack of better options. We can see many of the films grouped with films from the same genre, even though there is some mixing of the genres. We expected this as films belong to multiple genres, and even two films belonging to the same genre can be quite different.

4.3 Clustering Movies in the Guinness Sample

For the third sample, *Guinness Categories*, we decided again to look only at the cases with *cosine* as distance, and leave out the *euclidean* distance cases. The dendrograms can be seen in Figure 10 for the *cosine–complete* case and in Figure 11 for the *cosine–ward* case. The dendrograms are similar in both instances. For example, both have been able to classify all Swashbuckler films, except *The Princess Bride*, into one group. All horror films are in one group even though those groups have some extra films in them. And the three disaster films *The Towering Inferno*, *The Poseidon Adventure*, and *Armageddon* are in one group. Both have also been able to make a group of crime movies even though they have some differences in them. In *cosine–complete* this group has *The Godfather*, *Goodfellas*, *The Untouchables*, *Pulp Fiction* and *Con Air* (Figure 10). The *cosine–ward* instance (Figure 11) is mostly the same but *The Untouchables* have been changed to *A Clockwork Orange*.

There are also some films that neither case was able to cluster well. These include, for example, *Easy Rider*, which is classified next to horror movies, *King Kong* together with *Three Days of the Condor* and *Monkey Business* together with *Scarface*. All of these pairings have a relatively small number of tropes (see Appendix A), and even though the films in them have other more similar films in the sample, those films have multiple films that are better matches for them to connect to.

There is no clear winner among the tried methods for this sample. For example the film pair *Titanic* and *Now Voyager* are next to *Superman* in *cosine–ward* (Figure 11) and next to *Gone With the Wind* and *Casablanca* in *cosine–complete* (Figure 10). Then again, the film *On the Waterfront* is grouped with the horror movies in *cosine–complete* while it is in the same group as *Citizen Kane* and *Airplane* in the *cosine–ward* case. *Cosine–ward* seems to again make deeper trees than *cosine–complete*. For example the disaster film pair *Twister* and *Airport* are in their own group in the *complete* case but the films belong to the same group with the three other disaster movies in the *ward* case. All in all we think the *cosine–ward* case performs slightly better with this sample.

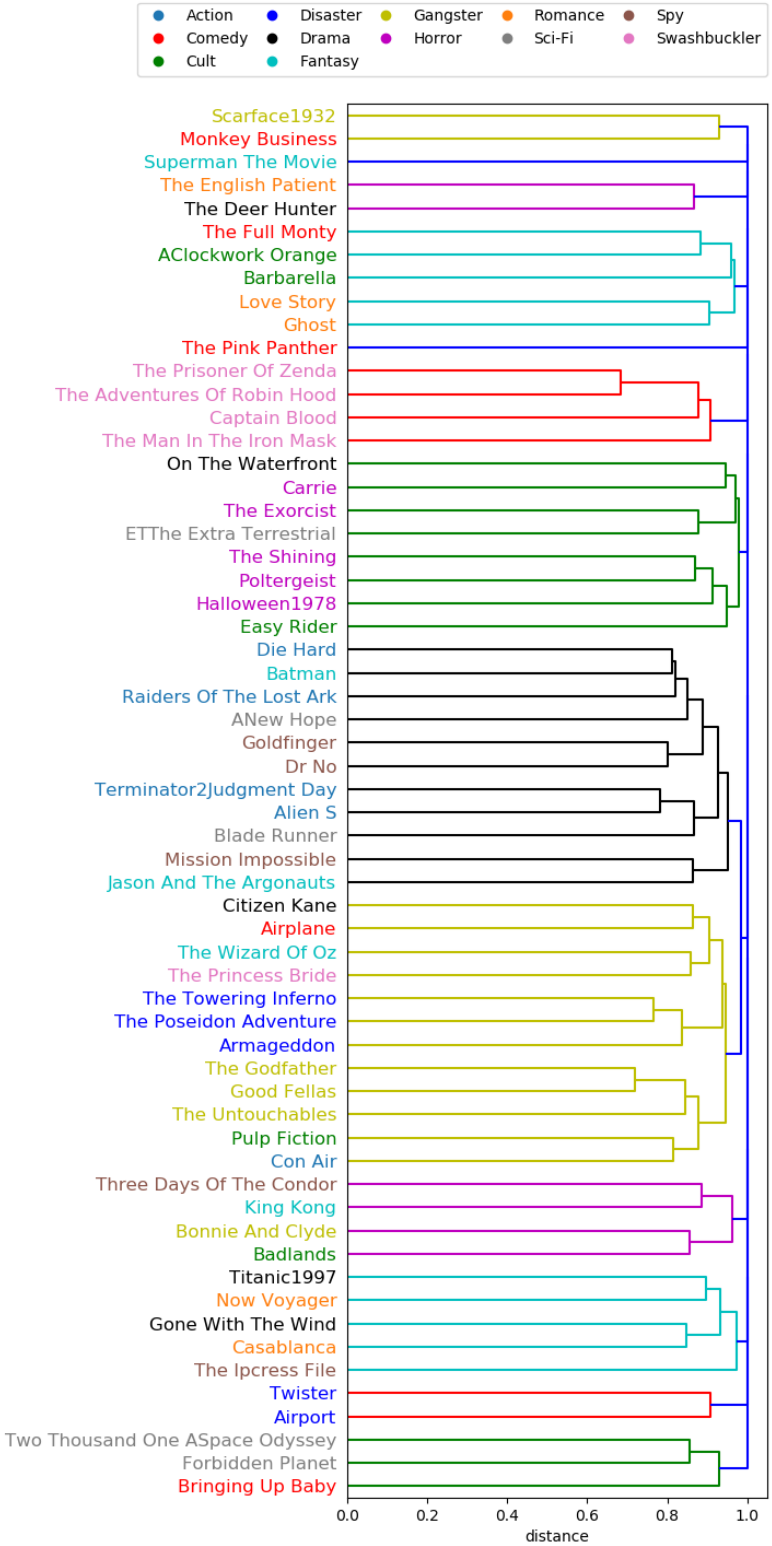

Figure 10: Dendrogram for sample *Guinness categories* using the *cosine* distance and *complete* linkage. The legend and the film name colors tells from which genre the films were taken.

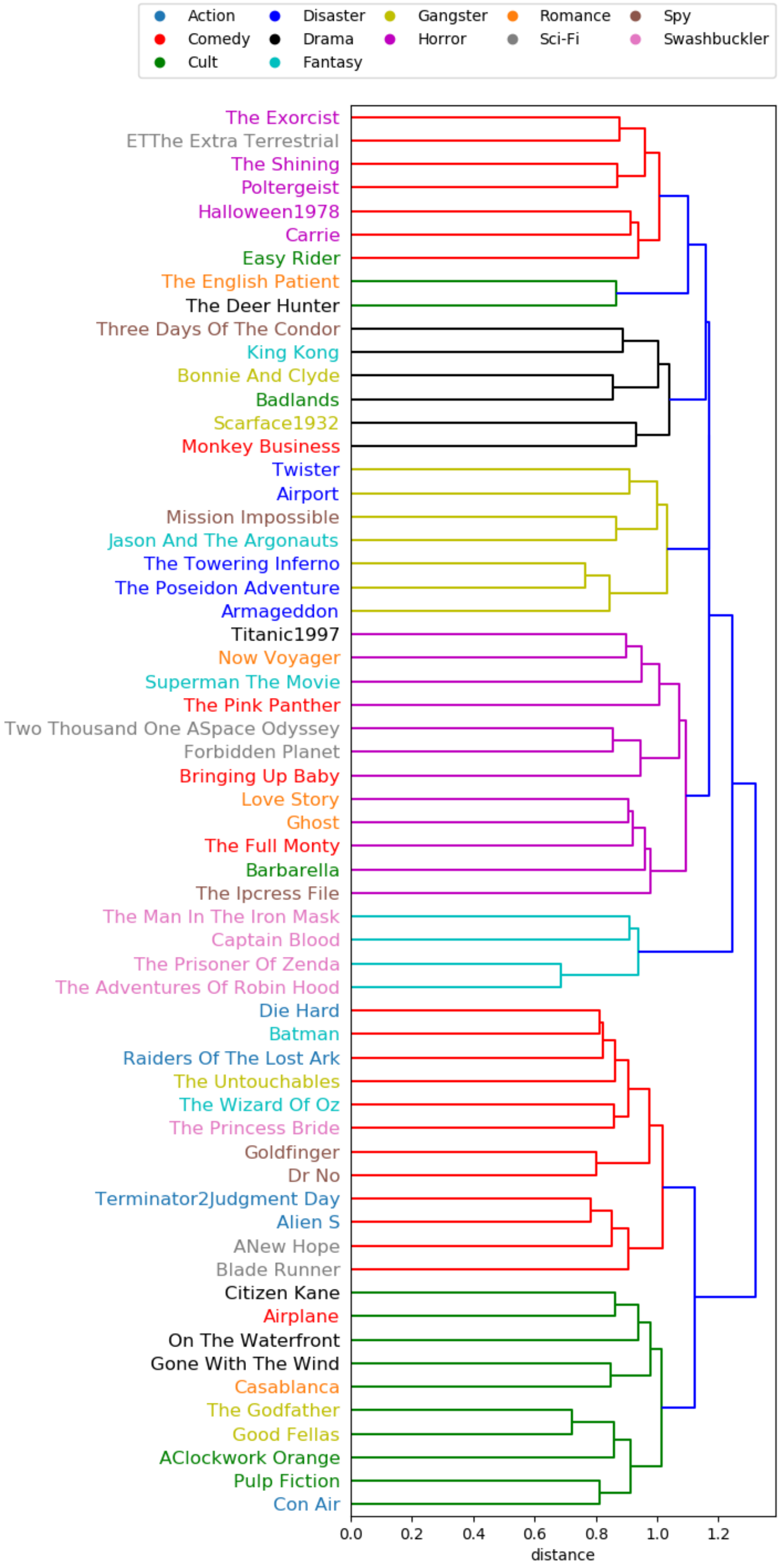

Figure 11: Dendrogram for sample *Guinness categories* using the *cosine* distance and *ward* linkage. The legend and the film name colors tells from which genre the films were taken.

4.4 Clustering Tropes in the Most Films Sample

In the last sample, *Most Films*, we look at clustering tropes instead of films using the same methods as before (Chapter 3). Short descriptions of the tropes can be found in Appendix B. We will again concentrate on the method combinations *cosine–complete* (Figure 12) and *cosine–ward* (Figure 13).

The clustering seems to work as well for tropes as it did for films. There are some trope pairs that both methods were able to cluster together. For example *Jerkass* is clustered together with *Jerk With a Heart of Gold*,*The Dragon* is clustered together with *Big Bad* and *Downer Ending* is clustered with *Driven to Suicide*. There were also tropes that both the methods have problems with. For example *Hey it's That Guy* and *Box Office Bomb* are clustered next to each other in both instances even though they do not fit together. This clustering happens however on a relatively large distance, which indicates that they are not that similar. Other tropes that we think do not fit were they were clustered are for example; *Precision F-Strike*, *Californian Doubling*, *Shout Out* and *Product Placement*. It is, however, hard to say where these would fit in an optimal clustering.

Again, there are differences in how the two methods cluster some of the tropes, and the differences get more visible when the clusters grow and distances become larger. One example of differences is that we have a cluster containing five violence-related tropes, in the *ward* case; *Too Dumb to Live*, *Gory Discretion Shot*, *Kick the Dog*, *Ax Crazy* and *Asshole Victim*, next to a group of two evil-related tropes; *Even Evil has Standards* and *Affably Evil*. In the *complete case*, *Ax Crazy* is also beside *Asshole Victim* and they are next to *Kick the Dog* and *Karma Houdini*, which sounds like a sensible cluster. However, *The Reveal* is between these and the Evil cluster. The tropes *Too Dumb to Live* and *Gory Discretion Shot* are their own group and not next to these other violence-related tropes.

Another difference can be seen in the order of how the pair *The Dragon* and *Big Bad* is clustered with *Badass*, *Action Girl* and *Big Damn Heroes*. We think these fit well together as the *Big Bad* is usually defeated by some kind of a hero in films. The difference is that in the *complete* case these are connected in the above-mentioned order, while in the *ward* case the tropes *Badass*, *Action Girl* and *Big Damn Heroes* form first their own cluster which is then connected to the *Big Bad's* cluster. We think that the latter of these two is more correct.

All in all, we think both methods, *complete* and *ward* works well for this scenario. Although we do feel that the *cosine–ward* case is slightly more easily interpreted than the *cosine–complete* case.

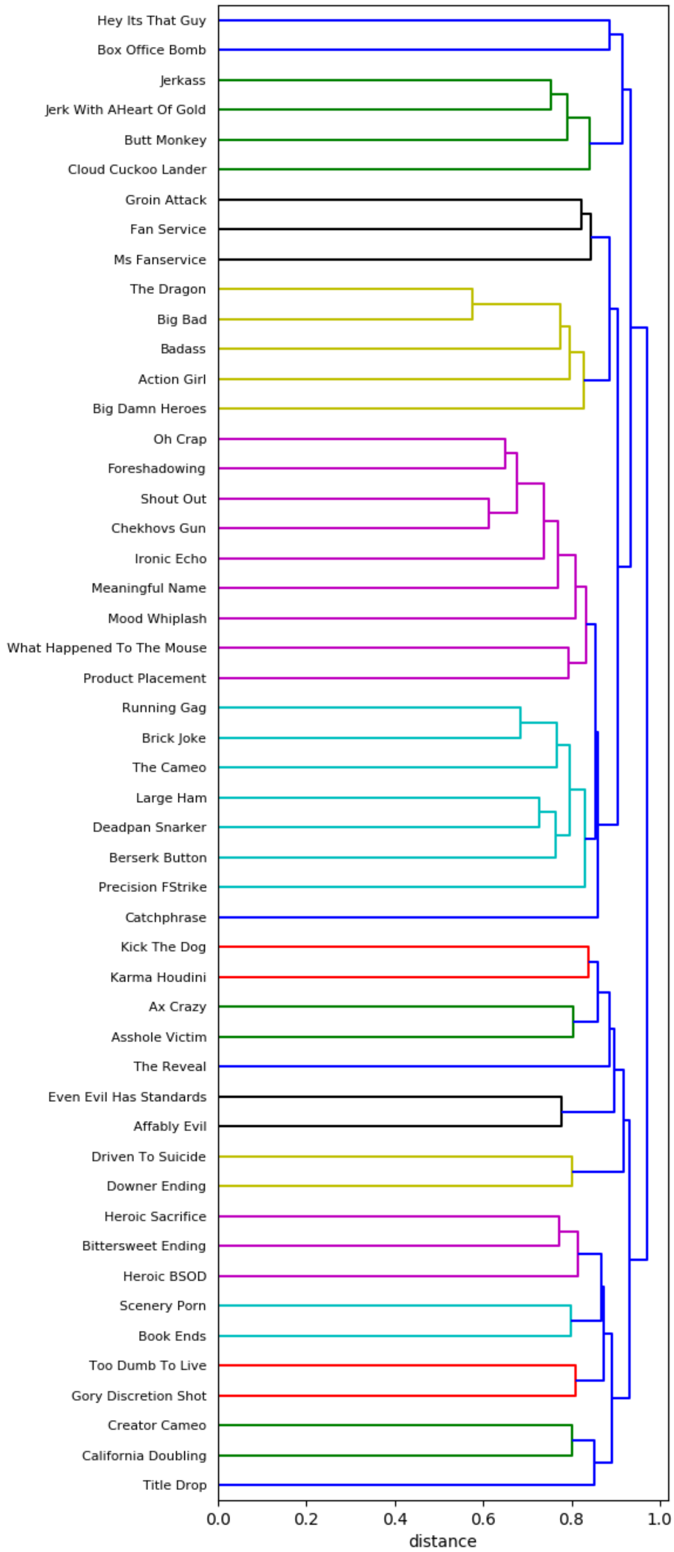

Figure 12: Dendrogram for the sample *Most Films* using the *cosine* distance and *complete* linkage. Here we cluster tropes with the help of their film connections.

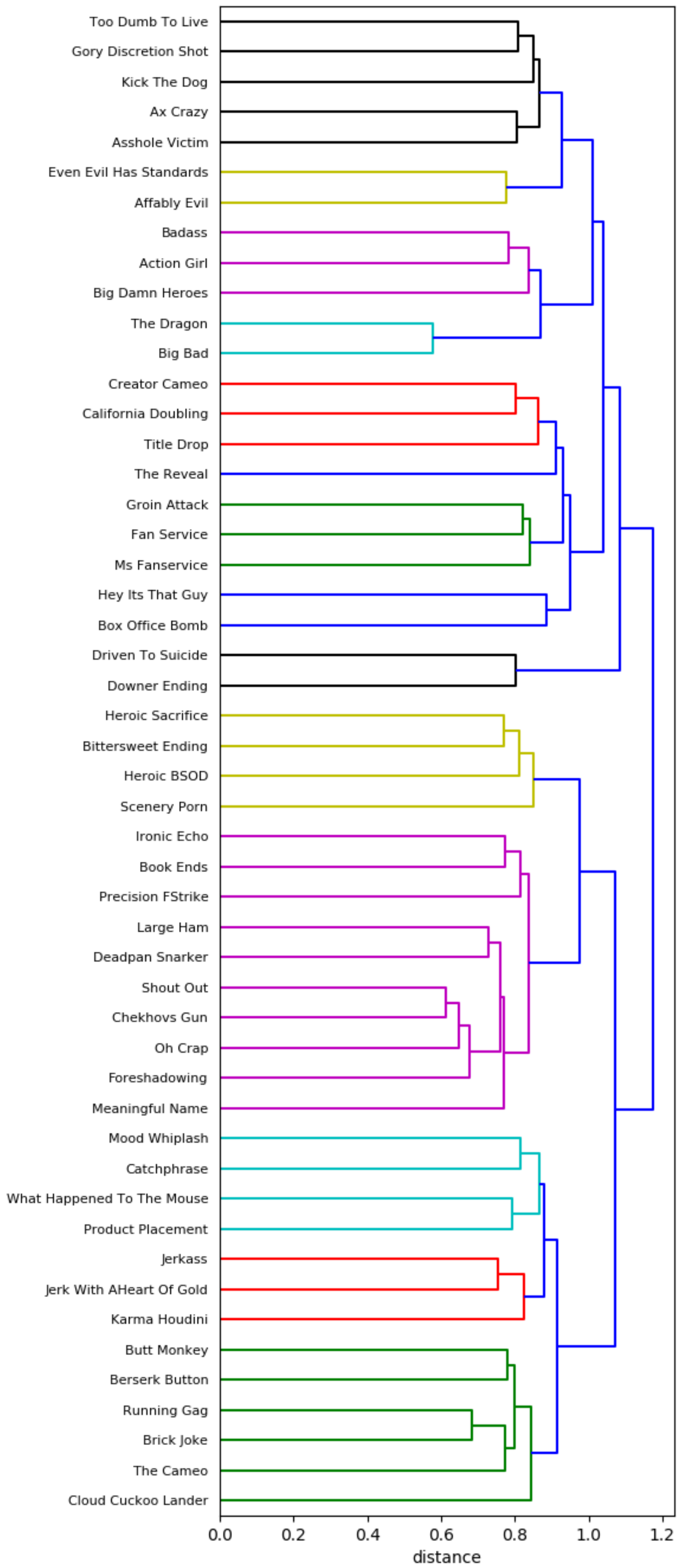

Figure 13: Dendrogram for the sample *Most Films* using the *cosine* distance and *ward* linkage. Here we cluster tropes with the help of their film connections.

5 Conclusions

In this project, we extracted film–trope connection data from the DBTropes RDF dataset [9], and the aim was to see if these connections could be used to cluster similar films or similar tropes. To do this we used hierarchical agglomerative clustering on three film samples and one trope sample.

The approach works well for clustering similar films or similar tropes. The *cosine* distance worked best for calculating distances between films from the film–trope connections. In the hierarchical clustering the *complete* linkage function, where which clusters are closest to each other is calculated from the farthest points from each other in the cluster, resulted in a more clearly defined dendrogram in the first, *Most Tropes* sample. In the three other samples; *IMDB Categories*, *Guinness*, and the trope clustering sample *Most Films*, the clearer clusters were obtained using the *ward* linkage function.

We used only small samples of the data, as our main goal was to see if clustering the films and tropes produce interpretable results. It would be interesting to see how well the clustering works when using a significantly larger sample. Also, testing other clustering or classification algorithms on this data is a promising line of future work. In the future the data could also be combined with other data about the movies, such as the publication year, IMDB rating, and movie description to get even better clustering results. It would also be interesting to look at the new tropescraper data by García-Ortega et. al. [7] to get more up-to-date data, as the DBTropes-dataset is from 2016.

References

[1] *Command-Line SPARQL with Jena*. http://richard.cyganiak.de/blog/wp-content/uploads/2013/09/jena-sparql-cli-v1.pdf.

[2] *Trope*, accessed 21.1.2020. https://tvtropes.org/pmwiki/pmwiki.php/Main/Trope.

[3] *DBTropes*, accessed 2.12.2019. dbtropes.org.

[4] *The top 100 films by genre type (top 5 films per genre) by guinness*, accessed 4.12.2019. https://www.filmsite.org/guinness.html.

[5] Datta, A., Kovaleva, S., Mardziel, P., and Sen, S.: *Latent factor interpretations for collaborative filtering*, 2017. https://arxiv.org/abs/1711.10816.

[6] García-Ortega, R.., Merelo-Guervós, J., García-Sánchez, P., and Pitaru, G.: *Overview of pictropes, a film trope dataset*, 2018. https://arxiv.org/abs/1809.10959.

[7] García-Ortega, R., García-Sánchez, P., and Merelo-Guervós, J.: *Startroper, a film trope rating optimizer using machine learning and evolutionary algorithms.* Expert Systems, 2020. https://onlinelibrary-wiley-com.libproxy.helsinki.fi/doi/full/10.1111/exsy.12525.

[8] Harper, F. Maxwell and Konstan, Joseph A.: *The movielens datasets: History and context.* TiiS, 5:19:1–19:19, 2015.

[9] Kiesel, Malte and Grimnes, Gunnar Aastrand: *Dbtropes—a linked data wrapper approach incorporating community feedback.* In Völker, Johanna and Corcho, Oscar (editors): *EKAW 2010 Demo & Poster Abstracts. International Conference on Knowledge Engineering and Knowledge Management (EKAW-10), 17th, October 11-15, Lisbon, Portugal*, October 2010. https://www.dfki.de/en/web/research/projects-and-publications/publications/publication/5070/, Best Poster.

[10] Kiesel, Malte and Hamed, Injy: *Goal-oriented annotation recommenders.* May 2013. https://www.researchgate.net/publication/329040864_Goal-Oriented_Annotation_Recommenders.

[11] Kiesel, Malte and Mittag, Florian: *Personalization in skipforward, an ontology-based distributed annotation system.* 781, January 2011. https://www.researchgate.net/publication/267713930_Personalization_in_Skipforward_an_Ontology-Based_Distributed_Annotation_System.

[12] Müllner, Daniel: *Modern hierarchical, agglomerative clustering algorithms*, 2011. https://arxiv.org/pdf/1109.2378v1.pdf.

[13] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E.: *Scikit-learn: Machine learning in Python.* Journal of Machine Learning Research, 12:2825–2830, 2011. http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf.

[14] Peska, Ladislav: *Linking content information with bayesian personalized ranking via multiple content alignments.* HT '17: Proceedings of the 28th ACM Conference on Hypertext and Social Media, pages 175 – 183, July 2017. https://dl.acm.org/doi/abs/10.1145/3078714.3078732.

[15] Peska, Ladislav: *Hybrid recommendations by content-aligned bayesian personalized ranking.* New Review of Hypermedia and Multimedia, 24(2):88–109, 2018. https://doi.org/10.1080/13614568.2018.1489002.

[16] Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L.: *BPR: bayesian personalized ranking from implicit feedback.* Proceedings of the Twenty-Fifth Conference of Uncertanity in Artificial Intelegence, AUAI, 2009. http://arxiv.org/abs/1205.2618.

[17] W3C Recommendation 25 February 2014: *Rdf 1.1 n-triples*, accessed 27.1.2020. https://www.w3.org/TR/n-triples/.

[18] Yang, Rui, Hu, Wei, and Qu, Yuzhong: *Using semantic technology to improve recommender systems based on slope one.* In Li, Juanzi, Qi, Guilin, Zhao, Dongyan, Nejdl, Wolfgang, and Zheng, Hai Tao (editors): *Semantic Web and Web Science*, pages 11–23, New York, NY, 2013. Springer New York, ISBN 978-1-4614-6880-6. https://link.springer.com/chapter/10.1007/978-1-4614-6880-6_2.

Appendices

A Sample Films

An alphabetic list of all the films in the 3 film samples described in Chapter 2.3. The table contains the film name, all categories from IMDB[6], and the number of tropes they have in the three data samples; *Most Tropes*, *IMDB Categories* and *Guinness*. In the category list, the name of the category is written with **bold** if that was the film category in the IMDB categories sample, and *cursive* if that was its category in the Guinness dataset. If a film has a category inside parenthesis that means that was its category in the Guinness dataset, but not on the IMDB webpage.

Film Name	All IMDB Categories	IMDB Categories	Guinness	Most Tropes
2001: A Space Odyssey	Adventure, *Sci-Fi*		145	
28 Days Later	Drama, **Horror**, Sci-Fi	155		
A Clockwork Orange	Crime, Drama, Scifi, *Cult*		127	
Airplane!	*Comedy*		141	
Airport	Action, Drama, Thriller, *(Disaster)*		8	
Alien	**Horror**, Sci-Fi	169		
Aliens	*Action*, Adventure, Sci-Fi		262	
Alladin	Animation, Adventure, Comedy			670
Amélie	Comedy, **Romance**	48		
Apocalypse Now	Drama, **Mystery**, War	176		
Armageddon	Action, Adventure, Sci-Fi, *(Disaster)*		115	
Austin Powers	Action, Adventure, Comedy			417
Avatar	**Action**, Adventure, Fantasy	267		403
Back to the Future	Adventure,**Comedy**, Sci-Fi	307		429
Badlands	Crime, Drama, *(Cult)*		19	
Barbarella	Adventure, Comedy, Fantasy, *(Cult)*		44	
Batman (1989)	Action, Adventure, *(Fantasy)*		274	381
Batman Begins	**Action**, Adventure	292		
Beauty and the beast	Animation, Family, Fantasy			593
Blade Runner	Action, *Sci-Fi*, Thriller		192	
Bonnie and Clyde	Action, Biography, Crime, *(Gangster)*		41	
Bringing Up Baby	*Comedy*, Family, Romance		36	
Captain America: The First Avenger	Action, Adventure, Sci-Fi			567

[6] `https://www.imdb.com/` (Accessed: 17.3.2020)

Film Name	All IMDB Categories	IMDB Categories	Guinness	Most Tropes
Captain America: The Winter Soldier	Action, Adventure, Sci-Fi			463
Captain Blood	Action, Adventure, *(Swashbuckler)*		27	
Carrie	*Horror*		15	
Casablanca	Drama, ***Romance***, War	144	138	
Citizen Kane	*Drama*, Mystery		139	
Clovefield	**Horror**, Sci-Fi, Thriller	121		
Con Air	*Action*, Crime, Thriller		143	
Deadpool	Action, Adventure, **Comedy**	4		
Die Hard	*Action*, Thriller		245	
Dr. No	Action, Adventure, Thriller, *(Spy)*		116	
E.T. the Extra - Terrestial	Family, *Sci-Fi*		82	
Easy Rider	Adventure, Drama, *(Cult)*		27	
Ethernal Sunshine of the Spotless Mind	Drama, **Romance**, Sci-Fi	65		
Finding Nemo	Animation, Adventure, **Comedy**	1		
Forbidden Planet	Action, Adventure, *Sci-Fi*		64	
Forrest Grump	Drama, **Romance**	181		
Frozen	Animation, Adventure, Comedy			752
Galaxy Quest	Adventure, Comedy, Scifi			374
Ghost	Drama, Fantasy, *Romance*		83	
Ghostbusters	Action, Comedy, Fantacy			442
Gladiator	**Action**, Adventure, Drama	159		
Goldfinger	Action, Adventure, Thriller, *(Spy)*		160	
Gone Girl	Drama, **Mystery**, Thriller	5		
Gone with the Wind	*Drama*, History, Romance		168	
Good Will Hunting	Drama, **Romance**	57		
Goodfellas	Biography, Crime, Drama, *(Gangster)*		173	
Groundhog Day	Comedy, Fantasy, **Romance**	117		
Guardians of the Galaxy	Action, Adventure, **Comedy**	425		564
Halloween	*Horror*, Thriller		79	
Harry Potter	Adventure, Family, Fantasy			390
Hellboy	Action, Fantasy, Horror			349
Hercules	Animation, Adventure, Comedy			448
Hot Fuzz	Action, Comedy, Mystery			374
Inception	**Action**, Adventure, Sci-Fi	318		398
Intouchables	Biography, **Comedy**, Drama	28		
James Bond	Action, Adventure, Thriller			892
Jason and the Argonauts	Action, Adventure, Family, *(Fantasy)*		75	
Kill Bill	Action, Crime, Thriller			451

Film Name	All IMDB Categories	IMDB Categories	Guinness	Most Tropes
King Kong	Adventure, Horror, Sci-Fi, *(Fantasy)*		54	
Life Is Beautiful	Comedy, Drama, **Romance**	31		
Love Story	Drama, *Romance*		22	
Memento	**Mystery**, Thriller	76		
Men In Black	Action, Adventure, Comedy			421
Mission Impossible	Action, Adventure, Thriller, *(Spy)*		106	
Monkey Bussiness	*Comedy*, Sci-Fi		8	
Monty Python and the Holy Grail	Adventure, Comedy, Fantasy			331
Mulan	Animation, Adventuyre, Family			413
Now You See Me	Crime, **Mystery**, Thriller	121		
Now, Voyager	Drama, *Romance*		18	
On the Waterfront	Crime, *Drama*, Thriller		22	
Pacific Rim	Action, Adventure, Sci-Fi			469
Poltergeist	*Horror*, Thriller		59	
Psycho	**Horror**, Mystery, Thriller	139		
Pulp Fiction	Crime, Drama, *(Cult)*		233	
Raiders of the Lost Ark	*Action*, Adventure		202	
Saw	**Horror**, Mystery, Thriller	2		
Scarface	Action, Crime, Drama, *(Gangster)*		26	
Se7en	Crime, Drama, **Mystery**	153		
Serenity	Action, Adventure, Sci-Fi			544
Shaun of the Dead	Comedy, **Horror**	216		
Sherlock Holmes	Action, Adventure, Crime			461
Shutter Island	**Mystery**, Thriller	20		
Slumdog Millionare	Drama, **Romance**	73		
Snatch	**Comedy**, Crime	134		
Spaceballs	Adventure, Comedy, Sci-Fi			362
Star Trek	Action, Adventure, Sci-Fi			636
Star Trek Into Darkness	Action, Adventure, Sci-Fi			421
Star Wars: Episode IV - A New Hope	**Action**, Adventure, Fantasy, *(Sci-Fi)*	259	226	
Star Wars: Episode V - The Empire Strikes Back	**Action**, Adventure, Fantasy	257		
Star Wars: Episode VI - Return of the Jedi	Action, Adventure, Fantasy			415
Superman	Action, Adventure, Drama, *(Fantasy)*		27	
Tangled	Animation, Adventure, Comedy			551
Terminator 2: Judgement Day	*Action*, Sci-Fi		261	380

Film Name	All IMDB Categories	IMDB Categories	Guinness	Most Tropes
The Adventures of Robin Hood	Action, Adventure, Romance, *(Swashbuckler)*		58	
The Avengers	**Action**, Adventure, Sci-Fi	209		
The Bourne Ultimatum	Action, **Mystery**, Thriller	21		
The Conjuring	**Horror**, Mystery, Thriller	34		
The Curious Case of Benjamin Button	Drama, Fantasy, **Romance**	42		
The Dark Knight	**Action**, Crime, Drama	534		676
The Dark Knight Rises	**Action**, Adventure	533		631
The Deer Hunter	*Drama*, War		37	
The English Patient	Drama, *Romance*, War		6	
The Exorcist	*Horror*		66	
The Full Monty	*Comedy*, Drama		21	
The Godfather	Crime, Drama, *(Gangster)*		332	411
The Hunchback of Notre Dame	Animation, Drama, Family			457
The Ipcress File	Thriller, *(Spy)*		11	
The Lion King	Animation, Adventure, Drama			634
The Little Mermaid	Animation, Family, Fantacy			447
The Lord of the Rings	Action, Adventure, Fantasy, Drama			827
The Man in the Iron Mask	Action, Adventure, Drama, *(Swashbuckler)*		40	
The Matrix	**Action**, Sci-Fi	399		578
The Pink Panther	*Comedy*, Crime, Romance		16	
The Poseidon Adventure	Action, Adventure, Drama, *(Disaster)*		73	
The Prestige	Drama, **Mystery**, Sci-Fi	98		
The Princess and the Frog	Animation, Adventure, Comedy			472
The Princess Bride	Adventure, Family, Fantasy, *(Swashbuckler)*		232	386
The Prisoner of Zenda	Action, Drama, Romance, *(Swashbuckler)*		39	
The Shining	Drama, ***Horror***	106	99	
The Sixth Sense	Drama, **Mystery**, Thriller	67		
The Three Stooges	Comedy, Family			315
The Towering Inferno	Action, Drama, Thriller, *(Disaster)*		42	
The Truman Show	**Comedy**, Drama, Sci-Fi	112		
The Untouchables	Crime, Drama, Thriller, *(Gangster)*		120	
The Unusual Suspects	Crime, **Mystery**, Thriller	153		
The Wizard of Oz	Adventure, Family, *Fantasy*		206	
Thor	Action, Adventure, Fantasy			577
Three Days of the Condor	Mystery, Thriller, *(Spy)*		36	
Titanic	*Drama*, **Romance**	130	131	

Film Name	All IMDB Categories	IMDB Categories	Guinness	Most Tropes
Toy Story	Animation, Adventure, **Comedy**	2		
Toy Story 3	Animation, Adventure, **Comedy**	2		
Transformers	Action, Adventure, Sci-Fi			396
TRON: Legacy	Action, Adventure, Sci-Fi			428
Twister	Action, Adventure, Thriller, *(Disaster)*		60	
Up	Animation, Adventure, **Comedy**	1		
Who Framed Roger Rabbit	Action, Adventure, Comedy			537
World War Z	Action, Adventure, **Horror**	119		
Wreck It Ralph	Animation, Adventure, Comedy			644
X-Men	Action, Adventure, Sci-Fi			580
X-Men: Days of Future Past	Action, Adventure, Sci-Fi			549
X-Men: First Class	Action, Adventure, Sci-Fi			548
Zombieland	Adventure, Comedy, **Horror**	190		

B Trope Info

A list of all the tropes, and their short descriptions, in the *Most Films* sample. The descriptions are either citations or shortened versions from descriptions from the Tv Tropes pages of the respective tropes. Accessed 14.4.2020.

Trope	Description
Action Girl	Female Badass.
Affably Evil	Seems like a normal polite person, except for the fact they are evil.
Asshole Victim	A character that is such a jerkass, you don't feel sympathy when horrible things happen to them.
Ax Crazy	A character who is psychologically unstable and presents a clear and present danger to others.
Badass	Character that possesses great prowess in physical fighting.
Berserk Button	Character always gets enraged when given a certain minor trigger.
Big Bad	The cause of all bad happenings in a story.
Big Damn Heroes	Any time the heroes/anti-heroes get to save the day in a big, awesome manner.
Bittersweet Ending	Somewhere between the Happily Ever After and the Downer Ending, the Bittersweet Ending happens when victory came at a harsh price.
Book Ends	Matching scenes at the beginning and end of a story.
Box Office Bomb	A movie for which production and marketing cost greatly exceeds its gross revenue.
Brick Joke	Named after an old joke, which seems at first blush to be a pair of unrelated jokes. At the end of the first joke, a brick is tossed away, leaving the confused listener without a punchline. At the end of the second joke, the brick returns and the listener falls on the floor laughing.

Trope	Description
Butt Monkey	The character who is always the butt of the joke or the "put them through hell" plotline.
Californian Doubling	A exotic foreign locale in a TV series or film is, due to budget or danger to the cast, likely to actually be somewhere in California.
Catchphrase	An expression used by a character numerous times.
Chekhov's Gun	An insignificant object that later turns out to be important.
Cloud Cuckoo Loader	A character with their head in the clouds. They are strangely oblivious to things that everyone else takes for granted.
Creator Cameo	When e.g. the film's director or producer appears in the film itself.
Deadpan Snarker	A character prone to gnomic, sarcastic, sometimes bitter, occasionally whimsical asides.
Downer Ending	A finale to a movie, or some other form of media that ends things in a way that varies from sad to tragic.
Driven to Suicide	Whatever the reason, a character may be driven to commit suicide.
Even Evil Has Standards	One of the easiest ways to highlight just how bad something or someone evil is: have an otherwise-remorseless villain reject it.
Fanservice	The use of sex or sexualized situations to reward or entice viewers.
Foreshadowing	A clue or allusion embedded in the narrative that predicts some later event or revelation.
Gory Discretion Shot	Blood or brains are seen splattering against a wall and the rest is left to the imagination.
Groin Attack	A character gets struck in the crotch, which is usually perceived as funny.
Hey Its that Guy	When you recognize a character's face and voice from another TV show.
Heroic BSoD	A stunning revelation or horrible event affects a character or someone they care deeply about, leaving them shocked to the point of mentally shutting down for a while.
Heroic Sacrifice	A character saves another/others from harm and is killed, crippled, or maimed as a result.
Ironic Echo	Character A makes a comment in the presence of Character B. Much later on, Character B makes the same comment in a different situation, often giving it an unexpected meaning in the new context.
Jerk With a Heart of Gold	A person you would expect to be a big Jerkass has some redeeming qualities behind their tough demeanor.
Jerkass	A Jerkass is a character who is so completely obnoxious that it is unbelievable anyone would willingly interact with them.
Karma Houdini	The character has done a number of things that deserve a karmic comeuppance, most importantly things that caused harm to the innocent. But when the time comes they don't get what they deserve. Instead, they get away scot-free.
Kick the Dog	When a character does something evil, cruel or very mean for no apparent gain, because the author wants to demonstrate that he's not a nice guy.
Large Ham	A ridiculously larger-than-life character, public speaker, or orator full of charisma, energy, and poetically powerful speeches.
Meaningful Name	A name that has a direct, barely-hidden meaning to it.

Trope	Description
Mood Whiplash	When tragedy and comedy comes close together.
Ms. Fanservice	A female character who provides a significant amount of Fanservice.
Oh Crap	The moment at which a character realizes that something really, REALLY bad is about to happen.
Precision F-Strike	A Precision F-Strike is when a character suddenly swears in an uncharacteristically strong manner.
Product Placement	The practice of prominently displaying or talking about a recognizable product in a film, in exchange for some consideration from the manufacturer, usually monetary.
Running Gag	A joke whose humor derives from repetition, ideally becoming funnier each time it is repeated.
Scenery Porn	Scenery porn is the emphasis on luscious backgrounds with great detail, lovely lighting or both.
Shout Out	A Shout-Out is something (a name, line of dialogue, or prop) in a show that refers to fans or family members of the cast or crew, or to another source of inspiration.
The Cameo	A brief appearance by a well-known star or another famous person in a role that would otherwise be an extra or walk-on.
The Dragon	The Dragon is the Evil Counterpart of the Number Two, being the right-hand man of the Big Bad.
The Reveal	The pivot in many plotlines is the Reveal. A character is revealed as another character's father, a god, or a secret suitor or arch nemesis in disguise. More broadly, the audience is given new information which had been withheld to create suspense.
Title Drop	If a line of dialogue is the title of the episode, movie, or book, it obviously must have some great significance.
Too Dumb to Live	Character who are so stupid that they simply do not deserve to survive.
What Happened to the Mouse	Occurs when a minor character, action, or very minor plotline is suddenly dropped from the story for no apparent reason, without any real explanation about what happened to it, and without a resolution.